

# TESTS — Autoevaluaciones

---



Profesor: Álvaro García Gutierrez



Acceso a Datos



Autoevaluación Diciembre— 10/12/2025



Resultado: 20/20

---



## Base de Datos con Java:

### Ficheros binarios y XML

---

1

¿Qué clase se usa para escribir bytes en un fichero binario?

#### Posibilidades:

- a) FileOutputStream
- b) FileWriter
- c) DataReader
- d) FileInputStream



Respuesta correcta: a)



Justificación:

`FileOutputStream` permite **escribir bytes** directamente en un fichero binario.

`FileWriter` trabaja con **caracteres**, no con bytes.

---

2

Fichero binario de registros de 48 bytes. Ir al tercer registro (índice 2)

#### Posibilidades:

- a) `seek(2 48L)`
- b) `seek(length())`
- c) `seek(3 48L)`
- d) `seek(48L)`



Respuesta correcta: a)



Justificación:

El offset se calcula como:

$\text{índice} \times \text{tamaño\_registro} \rightarrow 2 \times 48 = 96.$

---

3

¿Qué método de `FileReader` indica fin de fichero?

#### Posibilidades:

- a) `readLine()` devuelve cadena vacía
- b) `close()` devuelve false
- c) `available()` devuelve 0
- d) `read()` devuelve -1

✓ Respuesta correcta: d)

Justificación:

En Java, `read()` devuelve **-1** cuando se alcanza el final del fichero.

---

## 4 Evento SAX al encontrar texto dentro de una etiqueta

Posibilidades:

- a) `characters()`
- b) `endDocument()`
- c) `startElement()`
- d) `startDocument()`

✓ Respuesta correcta: a)

Justificación:

`characters()` se ejecuta cuando el parser SAX encuentra **texto** entre etiquetas.

---

## 5 Tamaño de un registro binario

Datos:

- `int id` → 4 B
- `char[10] nombre` → 20 B
- `int dept` → 4 B
- `double salario` → 8 B

Posibilidades:

- a) 40 bytes
- b) 48 bytes
- c) 28 bytes
- d) 36 bytes

✓ Respuesta correcta: d)

Justificación:

$4 + 20 + 4 + 8 = 36$  bytes

---

## 6 Primera línea de un XML bien formado

Posibilidades:

- a) `#include`
- b) `// XML file`
- c)
- d)

✓ Respuesta correcta: d)

Justificación:

La **declaración XML** indica versión y codificación.

---

## 7 Elemento principal de un XML

### Posibilidades:

- a) Nodo hoja
- b) Nodo raíz
- c) Nodo índice
- d) Nodo folio

✓ Respuesta correcta: b)

#### Justificación:

Todo XML bien formado tiene un único nodo raíz.

---

## 8 Modo de RandomAccessFile que crea el fichero y permite leer/escribir

### Posibilidades:

- a) "w"
- b) "r"
- c) "append"
- d) "rw"

✓ Respuesta correcta: d)

#### Justificación:

"rw" permite lectura y escritura y crea el fichero si no existe.

---

## 9 Fichero de 3600 bytes con registros de 36 bytes

### Posibilidades:

- a) 60
- b) 120
- c) 36
- d) 100

✓ Respuesta correcta: d)

#### Justificación:

$3600 / 36 = 100$  registros completos.

---

## 10 Bytes que ocupan int, double y char en Java

### Posibilidades:

- a) 8, 8 y 1
- b) 2, 8 y 1
- c) 4, 4 y 2
- d) 4, 8 y 2

✓ Respuesta correcta: d)

#### Justificación:

- int → 4 bytes
- double → 8 bytes
- char → 2 bytes (Unicode)

---

## 1 1 XML enorme, lectura secuencial una sola vez

### Posibilidades:

- a) SAX
- b) FileWriter
- c) DOM
- d) RandomAccessFile

✓ Respuesta correcta: a)

#### Justificación:

SAX procesa el XML **por eventos**, sin cargarlo entero en memoria. Ideal para ficheros grandes.

---

## 1 2 Método SAX donde se leen atributos

### Posibilidades:

- a) endDocument()
- b) error()
- c) startElement()
- d) characters()

✓ Respuesta correcta: c)

#### Justificación:

Los atributos se reciben en `startElement()`.

---

## 1 3 Resultado del FileWriter con append

### Código:

```
FileWriter fw = new FileWriter("contador.txt");
fw.write("1");
fw.close();

FileWriter fw2 = new FileWriter("contador.txt", true);
fw2.write("2");
fw2.close();
```

### Posibilidades:

- a) El texto 12
- b) Fichero vacío
- c) El texto 2
- d) El texto 1

✓ Respuesta correcta: a)

### Justificación:

El segundo `FileWriter` abre en **modo append**, añade al contenido existente.

---

## 1 4 ¿Qué es un fichero XML?

### Posibilidades:

- a) Fichero binario del SO
- b) Fichero de texto con etiquetas
- c) Formato solo de imágenes
- d) Exclusivo de BBDD relacionales

✓ Respuesta correcta: b)

### Justificación:

XML es un **formato de texto estructurado mediante etiquetas**.

---

## 1 5 API XML que carga todo en memoria

### Posibilidades:

- a) StAX
- b) JAXB
- c) DOM
- d) SAX

✓ Respuesta correcta: c)

### Justificación:

**DOM** construye un **árbol completo de nodos en memoria**.

⚠ **Nota:** no apto para XML muy grandes.

---

## 1 6 Desventaja de `FileReader` sin buffer

### Posibilidades:

- a) No soporta Unicode
- b) No funciona con rutas relativas
- c) Menor eficiencia
- d) Solo lee 1 KB

✓ Respuesta correcta: c)

### Justificación:

Leer carácter a carácter sin `BufferedReader` es **ineficiente**.

---

## 1 7 Método de `BufferedWriter` para salto de línea portable

### Posibilidades:

- a) `nextLine()`
- b) `writeLine()`

- c) `newLine()`
- d) `println()`

✓ **Respuesta correcta: c)**

📘 **Justificación:**

`newLine()` usa el separador de línea del sistema operativo.

---

## 1 8 Tipo de ruta: `src/main/resources/datos.txt`

**Posibilidades:**

- a) Ruta solo Linux
- b) Ruta URI
- c) Ruta absoluta Windows
- d) Ruta relativa

✓ **Respuesta correcta: d)**

📘 **Justificación:**

Es una **ruta relativa al directorio de trabajo**.

---

## 1 9 Offset para leer el registro n (36 bytes)

**Posibilidades:**

- a)  $n / 36$
- b)  $36 - n$
- c)  $n + 36$
- d)  $n * 36$

✓ **Respuesta correcta: d)**

📘 **Justificación:**

`offset = índice × tamaño_registro.`

---

## 2 0 ¿Qué hace `seek(long posición)`?

**Posibilidades:**

- a) Devuelve el tamaño
- b) Cierra el fichero
- c) Elimina un registro
- d) Mueve el puntero

✓ **Respuesta correcta: d)**

📘 **Justificación:**

`seek()` **desplaza el puntero de lectura/escritura** a la posición indicada.

---